A MILLENNIUM SURVIVAL GUIDE FOR IT PERSONNEL

OVERVIEW

Failure to resolve Millennium issues will:

- Compromise our commitment to the health and safety of our workers and the public
- Force generating plant shutdowns
- Impair our ability to deliver energy
- Adversely impact how we realize and account for revenue
- Create consequential liabilities

The *Millennium Survival Guide* is a document that provides the Application Developer with an understanding of the Year 2000(Y2K) date problem, and methods to resolve today's non-compliant code problems and methods to prevent non-compliant application development in the future.

The company has over 1500 applications. Each application has to be reviewed and a millennium strategy decision has to be made for each. Is the application acceptable as is? Does the application require coding modifications? Is the application obsolete? Will we replace the application? Does the application require a version upgrade?

A recommended approach is to first read the guide in its entirety. Then, depending on whether you are developing a new application, validating an application for Y2K compliance, converting a non-compliant application, replacing a non-compliant application, or upgrading a non-compliant vendor package, follow the appropriate steps outlined in the *Action List* section.

THE PROBLEM

The Year 2000 problem is easy enough to describe. Most computer systems represent dates in the format MMDDYY, where 12/31/95 represents December 31, 1995. The century is not represented in the date, and we simply assume that 12/31/95 refers to 12/31/1995. Most computer programs that perform arithmetic and logic operations on these date fields use only the last two digits of the year when they make their calculations. As long as all the dates in question are in the same century, this works fine. Problems arise, however, when the century changes. Subtracting 12/31/95 from 12/31/05 to determine someone's age, for example, does not produce the correct answer of 10. It actually produces a result of -90.

Although the problem is easy to describe, it is very difficult to solve for a number of reasons, and can be compared to looking for a needle in a haystack. The visual image of looking through hay is not difficult to conjure up, but the painstaking execution of the solution is awesome. The sheer size of the problem is the first of these. Dates are everywhere, which means that all program code must be examined to determine if a change is necessary. Utilities, like most large corporations, has thousands of programs containing millions of lines of code. A programmer will have to examine each of those lines and make a decision as to whether or not it has to be changed for the Year 2000. A date field can be called date, or it can be called ball game. Many people in the data processing industry, when confronted with the Year 2000 issue, refuse to believe the size or scope of the problem. Many of them argue that changing dates to include a century should be a relatively easy process. This fails to take into account the large number of changes that must be made, as well as, the coordination and testing of those changes. Ownership of the problem is critical to its solution.

DEFINITION OF MILLENNIUM COMPLIANT

The term, "**Millennium Compliant,**" is the quality of a system to provide all of the following functions:

- Handle date information before, during, and after January 1, 2000, including, but not limited to, accepting date input, providing date output, and performing calculations using dates or portions of dates

- Function accurately and without interruption before, during, and after January 1, 2000, without any change in operations associated with the advent of the new century

- Respond to two-digit year date input in a way that resolves the ambiguity as to century in a disclosed, defined, and predetermined manner

- Store and provide output of date information in ways that clearly define century

PURPOSE OF THE MILLENNIUM PROGRAM

The Millennium Program has been put in place to ensure against the unacceptable business consequences of computer systems failing as a direct result of millennium date incompatibility.

The Millennium Program has been put in place to protect and preserve investment in information technology by preventing significant computer system failures that would result from the inability of existing systems to accurately manage dates in the Year 2000 and beyond**.**

The Millennium Program will provide focus and consulting to business units in their efforts to fix non-IT equipment. IT equipment is any equipment that is under the maintenance and support accountability of any professional IT provider in the company or a contractor thereto.

The Millennium Program will provide focus, standards, program management, and resources to the IT community to fix all computer systems which they and the business unit system owners determine will fail as a direct result of millennium date incompatibility resulting in unacceptable business consequences.

The Millennium Program will seek out and require all computer system vendors to certify compliance of their systems in writing to the company, or, in the absence of such certification, will recommend a course of action to the appropriate managers.

The Millennium Program will budget all costs associated with enterprise level initiatives (e.g., awareness campaigns, outsourcing of work), as well as costs to analyze, define, design, test, implement, and verify compliance.

Costs associated with any end user labor resources needed to validate the business functionality of the systems will be budgeted by the business units.

Costs associated with fixing non-IT equipment will be budgeted by the business units.

WHO DO I CALL FOR HELP?

The Millennium Team is here to help. We welcome your questions, comments, suggestions, and ideas. We are all located at XXXX. Here is how to contact us:

	Internet Address	Telephone
Name	xxxxxx	XXXX

YEAR 2000 TESTING

Year 2000 testing requires that the Application Developer develop test cases primarily for input data testing of numerous conditions including leap year, date transaction validation, day/week/month in week/month/year calculations, data integrity, sequencing (i.e., JCL sort parameters, internal program sort), and time-sensitive data. In addition, every user must determine that his/her PC's system clock is Year 2000 compliant. Conditions at the Application Environment and Platform levels must be taken into account, as well.

The Software Millennium Test Development Guidelines Section should assist you in preparing these test cases:

- This section contains testing conditions that application developers must consider in preparing for YEAR 2000 changes.

- It also provides test conditions and associated date values–valid or invalid—for test cases especially applicable to unit testing. This is an ever-changing document and is updated and stored on the Lotus Notes *Millennium Document Library* under "Y2K Software Millennium Test Development Guidelines".

Conversion Methods

The Millennium Team is identifying and categorizing all applications by surveying application owners and compiling an application inventory. Application owners were able to identify their applications as applications that are required to be in service after the Year 2000, applications that are intended to be rewritten or replaced with a vendor package, or applications that are no longer necessary and considered obsolete.

All applications that are deemed required after the Year 2000 can be broadly categorized as either compliant (correctly processes date logic) or non-compliant (incorrectly processes date logic). These applications must be tested for compliance regardless of whether the application was originally categorized as compliant or non-compliant. From a high level perspective the following must occur for each application required to be in service after the Year 2000:

1. Identify the application as needed to be Year 2000 validated or modified to make it Year 2000 Compliant.

2. Develop and run a Year 2000 Test appropriate for the application.

3. Evaluate results. The process is complete if the Year 2000 Test proves that one of the following conditions is true:

   - *The application is compliant.*
     or
   - *The Year 2000 non-compliance is such that we can continue to use the application and do our work without the need for additional work. In other words, if the consequence of non-compliance is acceptable (i.e., a minor problem such as a report date or system that will only be non-compliant for a short period of time) the system may not be converted.*

The process will continue with the remaining steps only if the Year 2000 Test proves that all or a portion of the application needs further work

1. Run a Baseline Test—using an existing (modified, if necessary) or a new Baseline Test—to provide a reference to ensure that the functionality of the application has not changed after the code has been changed. This Baseline Test can be limited to only those portions of the code that have been changed.

2. Direct the programmers to make the Year 2000 coding changes.

3. Rerun both the Baseline Test and the Year 2000 Test to guarantee that there are no functionality changes and that the application is Year 2000 compliant.

The actual conversion (Step 5 above) can take place several different ways. We can convert an application with in-house resources, supplement in-house resources with contractors, or package the application and send it through the Conversion Factory that has been established. The Millennium Team will ensure a smooth conversion.

The Millennium Team has developed several detailed methodology templates for conversions and validations for both mainframe and client server applications. Each methodology addresses different situations by including different steps to follow to complete the conversion/validation. Each template has an associated checklist of detail procedures to follow for each of the steps. The following is a list of the currently developed templates and a brief description.

Detailed Methodology Templates

Mainframe Full Conversion Vendor with Baseline

A mainframe application will be converted by the selected conversion vendor off-site or on COMPANY premises. *(Template #1)*

Mainframe Full Conversion In-house With Baseline

A mainframe application will be converted by company personnel. *(Template #2)*

Mainframe Limited Conversion In-house Without Baseline

A mainframe application will be converted by company personnel. *(Template #3)*

Mainframe Validation With Baseline

A mainframe application will be validated for Year 2000 compliance by company personnel. *(Template #4)*

Mainframe Validation Without Baseline

A mainframe application will be assessed for Year 2000 compliance by the selected conversion vendor. *(Template #5)*

Mainframe Vendor Package Validation with Baseline

A mainframe application will be validated for Year 2000 compliance by company personnel. *(Template #6)*

Mainframe Vendor Package Validation Without Baseline

A mainframe application will be assessed for Year 2000 compliance by the selected conversion vendor. *(Template #7)*

Non-Mainframe Validation

A non-mainframe application will be validated for Year 2000 compliance by company personnel. *(Template #8)*

Non-Mainframe Conversion

A non-mainframe application will be converted by company personnel. (*Template #9)*

These templates can be found in the Lotus Notes *Millennium Document Library* under "Conversion Templates." If you do not have access to the Millennium Document Library or do not have access to Lotus Notes, please contact a representative of the Millennium Team for assistance.

Overview of the Conversion Options

**Once an application has been identified, and a methodology template has been chosen, the application developer must decide upon the specific technique to be used to bring each application program into compliance. The specific techniques include bridging, windowing, or date expansion,
or simply not convert. The following briefly describes each technique:**

Bridging

Bridging is the conversion method of choice if there are more than a few dates within a program. Bridging logic is added at the beginning of each program to expand the year to include the century. Bridging logic is also added at the end of each program to remove the century from the year. Therefore, data files coming in and going out of the program will remain in the same format. The century will be determined by a common program that will be accessed by each program in the application. Century will be determined by the following rule:

- If a year field is greater than 35 (e.g., “36” through “99”), “19” will be assumed to be the century. However, if the year is less than or equal to 35 (e.g., “00” through “35”), “20” will be assumed to be the century. This rule applies only if the application does not use dates prior to 1935.

The previously mentioned bridging process requires substantial setup such as cloning copybooks, creating new modules, and adding program logic containing a series of “Moves.” Thus, the bridging process is not efficient unless there are more than a few dates.

Windowing

Windowing is the conversion method of choice if there are only a few dates within a program. Instead of adding logic at the beginning and end of the program, Windowing logic is added following each date reference in the program. As in Bridging, each date is expanded to include the century. Again, data files coming in and going out of the program will remain in the same format. The century will not be determined by a common program; it will be determined by its own logic. The following rule will be used to determine the century:

- If a year field is greater than 35 (e.g., “36” through “99”), “19” will be assumed to be the century. However, if the year is less than or equal to 35 (e.g., “00” through “35”), “20” will be assumed to be the century. This rule applies only if the application does not use dates prior to 1935.

Date Expansion

Expanding the field in a file or column in a database is another conversion option. With this approach data will be physically expanded to reformat dates to a four digit year or other compliant format (DATE data type). *This is not a recommended method for mainframe applications.*

No Conversion

As a final option, company may choose not to convert an application that is non-compliant. Company may choose to accept a certain level of non-compliance if the consequence of non-compliance is acceptable (i.e., a minor problem exists, such as a report date or system that will only be non-compliant for a short period of time).

DATE STANDARDS RECOMMENDED TO BE YEAR 2000 COMPLIANT

As the need to exchange information across network boundaries increases, lack of common standard practices will become a formidable barrier to interoperability. It was identified that there are a variety of date standards being used within Company's IT complex. Depending on the environment and software/language or coding method, each application seems to maintain its own technique of expressing and storing date data. Many applications maintain different formats for input, output, display, and storage. Some have Julian formats, other have Gregorian formats, and even among those Julian and Gregorian formats, there are differences in representations (e.g., MMDDYY, YYMMDD, YYDDD). Some applications maintain numeric date formats *as binary, display, or packed* and others have alphanumeric representations. Applications sharing different date formats may be subject to additional risk of failure such as DATE data being distributed between different technologies (i.e., DB2, SYBASE) or downline feeds (i.e., Indus to other ad hoc applications).

For date input, report output, and screen displays, the USA standard date is to be utilized at COMPANY. This standard provides consistency for viewing and entering date data. The format of the USA Standard is:

Std	Name	Format	Length	Display
USA	**IBM USA Standard**	**MM/DD/YYYY**	**10**	**01/15/1996**

For data storage, most applications should use the current System date/time data type format supplied by their software. The advantage to this is that numbers representing dates and times can be stored in columns with numeric data types. Applications such as DB2 or SQL Servers (i.e., SYBASE) have the ability to recognize and load date or time values from outside sources, converting valid input values to their internal format. Another advantage is that they can store date and time information from January 1, 1753 through December 31, 9999.

There are some applications that cannot conform to a date data type format (i.e., ADABAS), and, therefore, should default to a character "8" ISO format (listed below). Although some COMPANY applications areas developed ADABAS systems in the past, the use of ADABAS is not the strategic direction for the company. In the future, as the larger ADABAS systems get replaced by packages and the smaller ones are converted into existing client server applications, the inconsistency between the two formats will become less of an issue. The format of the ISO Standard is:

Std	Name	Format	Length	Display
ISO	**International Standards Organization**	**YYYYMMDD**	**8**	**19960115**

ACTION LISTS

For any application developer at COMPANY, surviving the Y2K challenge will mean developing new applications that are Y2K compliant, validating existing applications for Y2K compliance, version upgrading for a non-compliant vendor package, replace with new vendor package or the actual conversion of applications for Y2K compliance. From a high level perspective, the following items should be performed for each unique application challenge.

New application development requires the following:

Understand the problem

Application developers must first understand the Y2K issues outlined in the beginning of the survival guide. Please refer to *Background, Magnitude of Problem,* and *Definition of Millennium Compliant* in the Survival Guide.

Understand Application Conditions

There are many things that an application can do that can cause non-compliance. Please refer to the *Application Conditions* section of the Survival Guide

Follow COMPANY date standards

The Millennium Team has developed display and storage date standards. Please refer to *Date Standards at COMPANY* section of the Survival Guide.

Develop new application using COMPANY date standards

Develop an application following all COMPANY standards and guidelines for new development including COMPANY date standards to ensure Y2K compliance.

Validate compliance using Y2K test plan

Develop a test plan that ensures all application transactions and conditions are Y2K compliant. Please refer to *Year 2000 Test Conditions* section of the Survival Guide or Step 4 Prepare Baseline/Y2K test cases in any methodology template identified in *COMPANY Conversion Method* section of the Survival Guide.

Version Upgrading of existing applications requires the following:

Understanding the problem

Application developers must first understand the Y2K issues outlined in the beginning of the survival guide. Please refer to *Background, Magnitude of Problem,* and *Definition of Millennium Compliant* in the Survival Guide.

Check for new version application conditions that may cause non-compliance

There are many things that an application can do that can cause non-compliance. Please refer to the *Application Conditions* section of the Survival Guide

Validate compliance using Y2K test plan

Develop a test plan that ensures all application transactions and conditions are Y2K compliant. Please refer to *Year 2000 Test Conditions* section of the Survival Guide or Step 4 Prepare Baseline/Y2K test cases in any methodology template identified in *COMPANY Conversion Method* section of the Survival Guide.

Replacing with new vendor applications requires the following:

Understanding the problem

Application developers must first understand the Y2K issues outlined in the beginning of the survival guide. Please refer to *Background, Magnitude of Problem,* and *Definition of Millennium Compliant* in the Survival Guide.

Ensure that the Millennium compliance language is in contract with vendor.

Company requires that all purchased and/or leased products meet date compliance requirements into and beyond the Year 2000 , with no interruption of service or additional expense. Any and all costs including, but not limited to, product upgrades and direct expenses incurred due to failures caused by the change in century, shall be the responsibility of the vendor**.**

If the product is, or will not be designed to meet Year 2000 compliance, the vendor must notify in writing prior to entering into any purchase agreement.

Check for application conditions that may cause non-compliance

There are many things that an application can do that can cause non-compliance. Please to the *Application Conditions* section of the Survival Guide.

Validate compliance using Y2K test plan

Develop a test plan that ensures all application transactions and conditions are Y2K compliant. Please refer to *Year 2000 Test Conditions* section of the Survival Guide or Step 4 Prepare Baseline/Y2K test cases in any methodology template identified in *Conversion Method* section of the Survival Guide.

Converting non-compliant applications requires the following:

Understand the problem

Application developers must first understand the Y2K issues outlined in the beginning of the survival guide. Please refer to *Background, Magnitude of Problem, and Definition of Millennium Compliant* in the Survival Guide.

Check for application conditions that may cause non-compliance

There are many things that an application can do that can cause non-compliance. Please refer to the *Application Conditions* section of the Survival Guide

Follow the recommended standards for conversion

The Millennium Team has identified several methodology templates for conversions and validations for both mainframe and client server applications. Please refer to the *Conversion Method* section of the Survival Guide to select the appropriate methodology template.

Validate compliance using Y2K test plan

Develop a test plan that ensures all application transactions and conditions are Y2K compliant. Please refer to *Year 2000 Test Conditions* section of the Survival Guide or Step 4 Prepare Baseline/Y2K test cases in any methodology template identified in *Conversion Method* section of the Survival Guide.

Y2K Millennium Project

Roles and Responsibilities

High Level Tasks	Y2K Team	IT Staff Managers	Software Owner
Planning			
1. Take ownership of the problem.		X	X
2. Validate for completeness the inventory of all applications.	X	X	X
3. Identify all developed application software.	X	X	X
4. Identify all vendor hardware and software.	X	X	X
5. Assume responsibility for a selected set of applications - Management Staff Responsibility (MSR) List		X	
6. Identify applications that are maintained by IT staff Managers.	X	X	X
7. Identify quality software/applications.	X	X	X
8. Initiate vendor Y2K compliance process.	X		
Scheduling			
1. Choose project conversion option.		X	X
2. Determine whether the work can be done by programming environment, or supplemented by the Y2K team resources. Find resources if staff is not available.	X	X	X
3. Identify/Commit/Coordinate resource to do the validation and/or conversion work.	X	X	X
4. Provide start date and a projected completion date for application to be validated or converted.		X	X
Conversion/Validation			
1. Provide Departmental Instructions for application testing or conversion.	X		
2. Develop a test plan for the applications for which you have responsibility.		X	X
3. Convert/Validate the application.	X	X	X
4. Test application for Y2K compliance.	X	X	X
Sign Off			
1. Sign off on the application indicating that it is obsolete, compliant, or ignored.		X	X

SOFTWARE MILLENNIUM TEST SIGNOFF

Software Title: _____ Revision No. ____

Application (if different from Software Title): _____ CR No._____

Software Owner, Title: _____

Prepared By, Title: _____

Check Appropriate Selection(s):

Millennium Testing Performed		**IF Millennium Testing *not* Performed**
___ Software is Millennium Compliant		___ Vendor certified software is millennium compliant. (Attach copy of vendor certification)
___ Software is ***not*** Millennium Compliant*		___ Software does ***not*** perform date input, output, or processing.
___ Conversion will be performed		___ Software can ***not*** be tested* - (reason is attached)
___ Conversion will ***not*** be performed*		___ Software is retired
		___ Software will be retired prior to date related problems*

*Contingency Plan is required to address actions if software conversion or retirement is ***not*** completed prior to date problems. The contingency plan must be attached to this document.

_____ _____
Software Owner/Computer Owner Date

_____ _____
Supervisor of Software Owner/Computer Owner Date

_____ _____
Manager of Department that Owns Software Date

SOFTWARE MILLENNIUM TEST DEVELOPMENT GUIDELINES

Software Title: _____ Revision No. ____

Application (if different from Software Title): _____

Software Owner, Title: _____

Prepared By, Title: _____

During the process of testing, apply a combination of verification and validation techniques. These techniques include:

1. Unit Testing
- 1.1. Testing the System Clock
- 1.2. Input Testing
- 1.3. Data Testing

2. System Testing
- 2.1. Stress Testing
- 2.2. Recovery Testing
- 2.3. Regression Testing
- 2.4. Error Handling Testing
- 2.5. Manual Support Testing
- 2.6. Parallel Testing

3. Integration Testing
- 3.1. Intra- and Inter-System Testing

4. PC Testing

The following sections will cover some useful testing techniques and scenarios for Year 2000 testing. They are not meant to be all inclusive. Therefore, it is important that additional tests be developed, as appropriate, for the application.

Attention: **By nature, Year 2000 exposures are time-sensitive and time-driven. Be cautious before resetting the system timer. Some system resources and functions are time-sensitive and may be activated or de-activated when the system clock is reset. Such effects can occur when the system clock is either set forward or backward. Without careful planning, you could cause the loss of these system resources and/or functions, some of which might contaminate the production system or production data bases when running various test scenarios.**

Unit Testing

Unit test is performed on one piece of software module at a time and is an exhaustive test of all logic within the module to demonstrate correctness and adherence to applicable specification and design requirements. Unit test should focus on exposing defects within the module logic (try to make it fail).

Testing the System Clock - This test involves resetting the system clock to identify problems which could occur (software, firmware, hardware, system access, etc.) when the century changes.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Expiration Test	Mainframe/Client Server	
			- User IDs		
			- Passwords		
			- Authorization/protection access		
			- Network access		
			- Automation functions		
			- SMS (System Managed Storage)/HSM (Hierarchical Storage Management) migrated data sets earlier than expected		
			2. Label driven tape datasets - are tapes expired earlier than expected? (i.e., validate label parameter expiration (99365, 99366))	Mainframe/Client Server	
			3. Archiving data expired earlier than expected?	Mainframe/Client Server	
			4. (12/31/1999 23:55 hrs) Monitor screen and transaction behavior	Mainframe/Client Server	

Unit Testing-Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			- Validate that dates are calculated and displayed correctly (i.e., 1999 rolls into 2000, ***not*** 1900)		
			5.Validate End of Processing logic to see if dates will be incorrectly interpreted and/or used.	Mainframe/Client Server	

Input Testing - Apply requirements testing to verify that the system performs its function correctly and that it remains functional over a continuous period of time.					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Will program respond correctly if “00” or “2000” is entered.	Mainframe/Client Server	
			2. Is a 4-digit year accepted or is it truncated?.	Mainframe/Client Server	
			3. Ensure xx/xx/xx date =xx/xx/xxxx after expansion or conversion for all databases and tables.		

Data Testing - Set the clock to test process cycles and automatic functions that are activated on a regular basis. These scenarios can be used to identify Year 2000 exposures that need to be fixed as well as to validate programs after applying Year 2000 solutions.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Leap year - Ensure that year 2000 is processed as a leap year.	Mainframe/Client Server	
			- 1996/2/29 should pass (1996 is a leap year)		
			- 2000/2/29 should pass (2000 is a leap year)		
			- 2004/2/29 should pass (2004 is a leap year)		
			2. Invalid Leap Year Test	Mainframe/Client Server	
			- 2/29/1999 non-leap year		
			- 2/29/2001 for non-leap year		
			3. Date Transaction Validation	Mainframe/Client Server	
			- (01/01/2000) Test processing for the first calendar day of the year		
			- (01/31/2000) Test and validate processing for the last business and calendar day of the month		
			4. Day-in-year calculation test	Mainframe/Client Server	
			- Does year 2000 have 366 days (not 365)?		

Data Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			5. Day-of-the-week calculation test	Mainframe/Client Server	
			- 02/28/2000 should be a Monday		
			- 03/01/2000 should be a Wednesday		
			- 01/03/2000 First business day of week		
			- 01/03/2000 First business day of month		
			- 01/03/2000 First business day of year		
			- 01/07/2000 Last business day of week		
			6. Week-of-the-year calculation test	Mainframe/Client Server	
			- The 11th week of the year 2000 is 3/5 to 3/11		
			7. End-of-Week Test	Mainframe/Client Server	
			- 01/08/2000 should be a Saturday		
			- 01/09/2000 should be a Sunday		

Data Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			8. Data Integrity	Mainframe/Client Server	
			- Are years 1800, 1900, 2000 distinguishable between one another?		
			- Validate for hard coded century occurrence of "19" and/or "20" in program code.		
			- Calculations - Look at programming logic to see if the usage of dates/date ranges in calculations will be correct		
			- Calculations - Check calculation when extends coverage into Year 2000 and verify future billing amounts are not impacted.		
			9. JCL/DCL CONTROL LANGUAGES	Mainframe/Client Server	
			- Ensure sorts use dates properly in processing		
			- Validate and test sort parameters		
			- Review sorts internal to programs		
			- Validate sort data sequence		
			- Record length adjusted - validate that increase records size are reflected in Record Length (LRECL - Logical RECord Length) field.		
			- Validate that Blocksize a multiple of LRECL		

Data Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			10. Age Test	Mainframe/Client Server	
			- Use 12/31/1899 to verify age and date of birth calculations		
			- Validate processing for roll-over to 2001		
			11. Time-sensitive data (may ***not*** be applicable to some applications)	Mainframe/Client Server	
			a.) Use current system clock and test data with dates:		
			- Before 01/01/2000		
			- After 01/01/2000		
			b.) Set system clock to 12/31/1999 and test data with dates:		
			- Before 01/01/2000 (12/15/1999) validate transaction calculations are correct within 10, 15, and 30 day period)		
			- After 01/01/2000 validate that everything behaves normally as 2000 approaches		

Data Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			c.) Set system clock after 01/01/2000 and test data with dates:		
			- Before 01/01/2000 validate backdated calculations are correct within 10, 15, and 30 day period)		
			- After 01/01/2000 Test - Set system clock to (02/29/2000) validate backdated calculations are correct within 45, 60 and 90 day period		
			- Set system clock to (03/31/2000) validate processing for the last business and calendar day of the quarter		
			- Set system clock to (03/31/2000) validate processing for the last business and calendar day of the quarter		

System Testing

System Testing ensures sufficient testing of a function's implementation and helps determine that all structures of the system are integrated to form a cohesive unit.

Stress Testing - apply stress testing to determine if the system can function when transaction volumes are larger than normally expected. The typical areas that are stressed include disk space, transaction speeds, output generation, computer capacity, and interaction with people. When testing Year 2000 changes, it is essential to verify that the existing resources can handle the normal and abnormal volumes of transactions after the restructuring of the code and the possible expansion of the data fields. For example, apply stress tests to determine:

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Can environment sufficiently accommodate the additional disk space required to support 2 to 4 digit expansion (DASD)?	Mainframe/Client Server	
			2. Are additional CPU cycles required to support code conversion (i.e., 2 digit encoding/ compression scheme) region size?	Mainframe/Client Server	
			3. Is response time adequate to support user turn around time?	Mainframe/Client Server	
			4. Do file definitions need to be reformatted (i.e., CI Splits, Data Dictionaries)?	Mainframe/Client Server	

Recovery Testing - Recovery Testing is used to ensure that the system can restart processing after losing system integrity. This is essential for systems in which the continuity of operation is critical to end users. Recovery processing normally involves the ability to go back to the last checkpoint, then reprocess up to the point of failure.
Can system restart processing after losing system integrity?

Any data integrity or unresolved exposures that lead to inconsistent data or code after you have implemented appropriate Year 2000 solutions will affect the completeness of backup data.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Can application go back to the last check point then reprocess up to the point of failure?	Mainframe/Client Server	
			2. Is documentation complete to support the manual manipulation of data?	Mainframe/Client Server	
			3. Can the system handle unconverted data (bridging available)?	Mainframe/Client Server	
			4. Verify results when a date is entered in one format (e.g. yymmdd/ccyymmdd) and displayed in a different format (e.g. mmddyy/mmddccyy). (2-byte-MF…4-byte-C/S format). Test for Julian dates, especially for calculations and Job Schedule Calendar.	Mainframe/Client Server	

Regression Testing- Ensures that all aspects of a system remain functionally correct after changes have been made to a program in the system. Because the potential exists for a tremendous amount of data and programs to be involved in your Year 2000 transaction, any change to an existing program in the system can have a snowballing or cascading effect on other areas in the system. A change that introduces new data or parameters, or an incorrectly implemented change can cause a problem in previously tested parts of the system, simply because of the way data can be shared between software entities.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Are user requirements followed (i.e., quality assurance)?	Mainframe/Client Server	
			2. Changes meet design specifications?	Mainframe/Client Server	
			3. Changes compliant with organization's policies and procedures?	Mainframe/Client Server	
			4. Validate data output records - data field following date field expansion.	Mainframe/Client Server	
			5. Validate data output records - data field in front of date field expansion.	Mainframe/Client Server	
			6. Validate on-line screen display field for error.	Mainframe/Client Server	
			7. Ensure all scheduling based on date return the same results before and after Y2K changes.	Mainframe/Client Server	
			8. Ensure conditions cover time zone differences.	Mainframe/Client Server	
			9. Ensure all extracting basedate returns the same results before and after Y2K changes.	Mainframe/Client Server	

Regression Testing- Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			10. Ensure all index processing based on date returns the same results before and after Y2K changes.	Mainframe/Client Server	
			11. Ensure all subscripting based on e returns the same results before and after Y2K changes.	Mainframe/Client Server	

Error Handling Testing - Determines if the system can properly process incorrect transactions that can be reasonably expected as types of error conditions. Error-handling testing is necessary to determine the ability of the system to properly process incorrect transactions that can be reasonably expected ass types of error conditions. For example, programs that accept only 4-digit year data entry format need to provide error messages for data entry in 2-digit year format, and vice versa for programs that accept only 2-digit year data entry format. When changing from 2-digit year format to 4-digit year format, you need to apply error-handling testing to verify the appropriate error-handling functions.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Normal error handling for current 4 digit year data entry when 2 digit data entry occurs.	Mainframe/Client Server	
			2. Normal error handling for current 2 digit year data entry when 4 digit data entry occurs.	Mainframe/Client Server	

Manual Support Testing - Evaluate the process by which the end user handles new data generated from the automated applications with Year 2000 support. Types of data from these applications include data entry and report generation. Any new data format should be easy to understand and ***not*** ambiguous. This method includes testing the interfaces (for example screens, procedures, operation manuals, and online HELP panels) between end users and the application

program. End users should be trained and use procedures provided by the system personnel. Testing should be conducted without any other assistance.					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Are new field on on-line screens ambiguous?	Mainframe/Client Server	
			2. Operation manuals updated with new procedures?	Mainframe/Client Server	
			3. On-line HELP panels updated?	Mainframe/Client Server	

Parallel Testing - Determine whether the processing and results of an application's new program version are consistent with old program version. Parallel testing requires that the same input data be run through the two versions of the application. However, if the new application changes data formats, such as reformatting the year-date notation to 4-digit format, you must modify test input data before testing.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Validation output report - Determine if data displays will be acceptable (compliant) in Year 2000	Mainframe/Client Server	
			- Date fields		
			- Non-date fields		
			- Report headers		
			- Report footers		
			2. Validate on-line screens - Determine if data displays will be acceptable (compliant) in Year 2000	Mainframe/Client Server	
			- Date fields		
			- Non-date fields		
			- Screen headers		
			- Screen footers		
			- On-line screen help		
			3. Validate that hard-coded dates or century indicators are ***not*** located in output records.	Mainframe/Client Server	

Parallel Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			4. Validate that hard-coded dates or century indicators are ***not*** located on output reports.	Mainframe/Client Server	
			5. Validate that hard-coded dates or century indicators are ***not*** located on screen displays.	Mainframe/Client Server	
			6. Validate output reports for zero suppression (i.e., year "00" would ***not*** display).	Mainframe/Client Server	
			7. Validate screen displays for zero suppression (i.e., year "00" would ***not*** display).	Mainframe/Client Server	
			8. Verify that the portion of the system that have *no* changes still runs properly as changes are made to other portions of the system.	Mainframe/Client Server	
			9. Verify that the program handles all its transactions correctly and remain stable for a defined period of time.	Mainframe/Client Server	
			10. Ensure that programs (or table subscripts) can handle date ranges that cross Millennium. - Some programs used dates as subscripts. (i.e., "00" for Year 2000 would be an invalid entry, September 1999 "999" my be considered as an end of file marker.)	Mainframe/Client Server	

Integration Testing

Intra- and Inter-System Testing - Applications are frequently connected with other applications to provide a higher or deeper level of functionality. Data may be shared between applications or systems. Intersystem testing is required to ensure that the connection functions properly between the applications. This test determines that the proper parameters and data are correctly passed between applications, and proper coordination and timing of each function exists between applications.

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Proper parameters passed between applications?	Mainframe/Client Server	
			2. Data transferred in the proper format to inter-system?	Mainframe/Client Server	
			3. Verify that the system can accept input from, and provide output to, other systems with which it interfaces as interfaces change.	Mainframe/Client Server	

PC Testing

Some older models of the PC may ***not*** have the capability to set or roll over the system clock beyond the year 2000 because the Basic Input/Output System (BIOS) is unaware of the century digit.
(Ex. ROMBIOS in most older PCs can ***not*** handle the year 2000 rollover correctly. Different versions of BIOS behave in different manners: Some roll the date to 1/4/00 or 3/1/00; Some roll the date to 1/1/80 or 3/1/80; Some just leave the date at 12/31/99)

Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			1. Test if the system clock can be set beyond the year 2000.	PC	
			- Set the system clock to 01/01/2000, reboot PC and recheck the date.		
			2. Test system clock automatic update function.	PC	
			a.) Test the system clock automatic update function when the power is on. - Set clock to 12/31/1999, 23:58:00, keep power on, validate date when clock reaches the year 2000. - Power off PC and recheck the DOS date.		
			b.) Test the system clock automatic update function when the power is off. - Set system clock to 12/31/1999, 23:58:00, power off PC and wait until the clock reaches the year 2000. - Power on PC and recheck the DOS date.		

PC Testing - Continued					
Test Applicable (☑ check if valid)	Compliant Yes	Compliant No	Test	Test Applies to:	Comments
			3. Test time update by the operating system	PC	
			a.) Update After Suspension of a time-sensitive program: - Set system clock to 12/31/1999, 23:58:00; suspend a time display program without a "wake up" timer; keep power on; wait until the clock reaches the year 2000; resume time display program; and check the date.		
			b.) Update After Suspension and Wake Up of time-sensitive program: - Set system clock to 12/31/1999, 23:58:00; suspend a time display program with a "wake up" timer set at 01/01/2000, 00:01:00; keep power on; wait until the time display program "wakes up"; check the date.		
			4. Leap Year Test	PC	
			a.) Change date 02/29/2000. If an error occurs, then BIOS is incorrect.		
			5. Test CPU	PC	
			a.) Use different machines (286/386, etc.) when executing tests to ensure processing time isn't impacted.		